Knowledge-Based Systems in the Era of Large Language Models: A Case Study in Fire Safety Management

Dilan Durmus [1], Shabtai Isaac[2], Alessandro Carbonari[1], Alberto Giretti[1]

[1] Polytechnic University of Marche, DICEA Department, Ancona, Italy

[2] Ben-Gurion University of the Negev, Department of Civil and Environmental Engineering, Beer-Sheva, Israel

d.durmus@pm.univpm.it,isaacsh@bgu.ac.il, alessandro.carbonari@staff.univpm.it, a.giretti@staff.univpm.it

Abstract –

The construction industry faces complex decision-making challenges that require the integration of extensive domain knowledge and data analysis. Traditional methods are inadequate to effectively address these challenges due to the dynamic nature of construction projects. In the fast-evolving landscape of construction technology, artificial intelligence (AI) plays a crucial role in enhancing decision-making and prediction processes. Therefore, the main purpose of this research is to examine how AI, particularly Large Language Models (LLMs), can be integrated in the analysis and management of emergency scenarios to improve decision-making, prediction, and optimization through knowledge-based systems. This research follows the CommonKADS method for knowledge engineering and embeds sub-symbolic and symbolic AI techniques. By integrating LLMs, this research conceptualizes and operationalizes tacit construction knowledge into structured knowledge systems. This approach has been applied in a case study analysing National Fire Protection Association (NFPA) fire incidents to test possible interactions to extract related data for the knowledge-based systems. The findings inspire future research on the scalability of AI-integrated systems across different segments of the construction industry, their potential in regulatory compliance scenarios, and the development of automated knowledge-based systems in construction industry.

Keywords –

Knowledge-based systems; Large Language Models; Fire Safety Management; Artificial Intelligence; Construction Industry

1 Introduction

The construction industry plays an important role in shaping the built environment but is facing challenges arising from the complexity and dynamic nature of its processes. These challenges are multifaceted, involving the coordination of multiple stakeholders, adapting to changing environmental conditions and complying with complex regulatory requirements especially in case of emergency management.

Traditional decision-making tools are often insufficient to meet the demands, as they struggle to proceed with large volumes of data and fail to pace up with rapid changes on construction projects. For example, Zhu et al. identifies challenges facing decision support systems in construction in the following aspects: impreciseness or uncertainty of information, dynamic and unstable situations, high volume, high velocity and high variety of the information.[1]

Algorithmic approaches to decision support show systemic limitations that push away the possibility of achieving optimal solutions for effective real time management in the era of big data. This creates a need for innovative approaches that are able to cope with result accuracy, that are computational, efficient, responsive and effective in the fast-paced world of the construction industry.

Recent developments in artificial intelligence (AI) technologies have produced significant advances to knowledge technologies, bringing renewed attention to the role of knowledge in solving complex problems. The introduction of knowledge-based systems (i.e. expert systems) in the 1980s marked a significant milestone in addressing the automation of knowledge-based problem solving. These systems used deep domain knowledge and heuristic reasoning to tackle complex, real-world problems. Their declarative nature allowed reasoning under uncertainty and provided explainable rational for their decisions [2], [3].

However, knowledge-based systems are not without limitations, such as the labor-intensive process of knowledge acquisition, scalability challenges, and an inability to provide creative, adaptive and context-specific responses[2]. Moreover, the cost of developing and maintaining such systems remains a significant barrier.

Knowledge engineering (KE) emerged to address these challenges by systematically structuring and organizing knowledge. KE methodologies, such as CommonKADS [4], provide robust frameworks for modeling tasks, agents, and knowledge systems. These frameworks emphasize conceptualization, operationalization, and formulation, in other words, knowledge coding, which ensure the systematic transformation of domain knowledge into structured, machine-readable and reusable systems [4]. While conceptualization involves identifying and organizing domain knowledge; operationalization translates abstract concepts into actionable procedures; and formalization encodes these procedures into formal representations that can be implemented in computational rule-based systems[4], [5]. As an example, CLIPS (C-Language Integrated Production System), a rule-based tool, integrates AI outputs into operational systems by providing components like a fact-list, knowledge base, and inference engine, enabling the application of domain-specific rules as per CommonKADS methodology [6].

Despite advances, traditional KE faces obstacles in integrating tacit knowledge (informal, experiential knowledge) into structured formats. This includes the significant resources required for knowledge extraction and representation, making it labor-intensive and costly. Additionally, adapting existing models to new circumstances is challenging, as traditional methodologies often lack the flexibility needed to accommodate evolving knowledge and dynamic environments effectively. The rise of artificial intelligence (AI), in particular the emergence of Large Language Models (LLMs), offers transformative opportunities to overcome these limitations.

LLMs such as GPT-4 are adept at processing unstructured data, identifying patterns, and generating human-like responses [7], [8], thereby operationalizing tacit knowledge and transforming natural language into structured representations. These capabilities, when combined with methodologies like CommonKADS, appears as a promising approach to automate and scale KE processes, creating adaptive, dynamic systems that bridge the gap between symbolic and sub-symbolic AI.

However, LLM systems show some severe limitations, due to their black-box architecture and statistical nature of the computing, which are ill-suited to the requirements of reliability and traceability of reasoning. A reliable and explainable reasoning process is in fact required in construction decision-making, because the responsibilities assumed by the decision maker are a determining factor in the decision-making process. Hence hallucination and the lack of explainable reasoning are a barrier to the straightforward application of AI in decision-making [9], [10], [11].

The idea claimed in this paper is therefore to use AI to improve the fragility of domain models and the high cost of building the knowledge base of symbolic systems on the one hand, and to define a procedure to mitigate the hallucination and explainability problems of these systems on the other.

Building upon our previous article presented at ISARC 2024 [12], where we introduced the AI-in-the-loop model as a novel approach for the construction industry, this paper serves as a direct follow-up. While the previous one is focusing on detecting and addressing the issue of hallucinations in AI outputs, this study further explores the integration of AI, particularly LLMs, into established knowledge engineering frameworks dynamically to mitigate such hallucinations more effectively.

Technically, this study aims to structure and operationalize tacit construction knowledge into dynamic, scalable systems by following the guidelines of CommonKADS methodology. The case study is centered on National Fire Protection Association (NFPA) fire incidents because fire safety management, a critical area in construction, provides a rich testing ground for these technologies, requiring fast, responsible decision-making under uncertainty [12], [13]. The research demonstrates how knowledge modelling strategy is done by an expert manually and then how LLMs could contribute to automate or speed up some steps by replacing the manual work done by the domain expert by extracting, structuring, and categorizing the elements in a ontological framework for improved safety management decisions.

This combines structured methodologies with the adaptive capabilities of LLMs to set the foundation for future research into scalable and reliable AI-driven systems across the construction industry.

The following sections will look into the integration of Large Language Models in knowledge-based systems. Section 2 will elaborate on the essential forms of knowledge and knowledge modelling, Section 3 will explain the integration of LLMs into KE. This is followed by a case study on fire safety management in Section 4 as a practical application of LLMs into KE processes. This paper will conclude with a short discussion.

2 Knowledge Modelling

Knowledge is a multi-dimensional asset that is critical to human reasoning and decision-making. In the field of knowledge engineering (KE), it is essential to understand and benefit from forms of knowledge to develop systems that simulate expert problem solving. Knowledge is commonly classified into explicit and tacit types, as well as declarative and procedural forms, each presenting unique opportunities and challenges for formalization and utilization in computational systems [4], [14].

Explicit knowledge refers to knowledge that can be expressed, codified, and shared, such as manuals, standards, and databases. It is relatively straightforward to capture and encode in knowledge-based systems (KBS) using tools like ontologies and knowledge graphs [5], [4]. In contrast, *tacit knowledge* is personal, context-specific, and often unspoken. This includes skills, intuitions, and experiences accumulated over time. This type of knowledge is more challenging to formalize, requiring advanced methodologies and tools for extraction and representation.

Declarative knowledge, or "know-what," captures phenomenological information and relationships, often suitable for logical reasoning and explicit modelling. *Procedural knowledge*, or "know-how," involves the steps or methods for performing tasks, making it valuable in systems that simulate decision-making processes [2], [4].

Methodologies such as CommonKADS provide structured frameworks for transforming tacit and procedural knowledge into explicit, actionable formats. This process consists of three phases: conceptualization (identifying and organizing knowledge), operationalization (structuring knowledge into workflows or rules), and formulation (formalizing it into computational representations) [4], [5].

Knowledge modelling is a cornerstone of knowledge engineering (KE), serving as the process by which knowledge is identified, structured, and formalized into models that are usable by computational systems. The strategy for knowledge modelling is essential for enabling effective decision-making, reasoning, and problem-solving in knowledge-based systems. At its core, knowledge modelling aims to bridge the gap between human expertise and computational implementation by systematically capturing and representing the subtleties of domain knowledge.

Knowledge models are abstract representations of the knowledge embedded in a domain. These models provide a structured framework to capture relationships, dependencies, and tasks, ensuring that knowledge is both understandable and applicable. The CommonKADS methodology outlines a series of models that captures the organizational, task, and agent aspects of a system, including a specialized knowledge model for domain-specific information and reasoning processes [4].

Following the CommonKADS approach, in this research, the knowledge base will be created in three stages:

1. **Conceptualization:** The initial phase involves identifying domain concepts and relationships. This stage captures both tacit and explicit knowledge through interviews, reports, and observations with domain experts. The conceptualization process prepares the ground for formal representation by structuring informal knowledge into a coherent mental model [4].
2. **Operationalization:** After conceptualization, the abstract model is translated into actionable workflows, algorithms, or rules that the system can execute. This step allows the system to mimic expert reasoning and decision-making processes by including input-output relationships, logical dependencies and procedural workflows [2], [5].
3. **Formalization:** The formalization phase is a straightforward translation of the operational knowledge arrangement into the language of the inference engine. This phase involves encoding the operationalized knowledge into formal computational languages, for example codes, or XML representations, such as ontologies, rule-based systems, or knowledge graphs. The ontological alignment can be considered as a complementary and optional phase linking the conceptualization performed with the ISO standards. This aims to enable reasoning and inference while ensuring consistency [4], [5].

However, it should be noted that one of the critical challenges in knowledge modelling is the representation and analysis of tacit knowledge. Experiential and context-specific tacit knowledge is difficult to formalize and integrate into structured systems. This requires intensive and strong expert knowledge, adaptability and serious time dedication.

3 Integrating Large Language Models into Knowledge Engineering

The integration of Large Language Models (LLMs) into knowledge engineering (KE) indicates a transformative shift in the field. LLMs, like GPT-4, are general-purpose technologies that highly useful in processing unstructured natural language, transforming it into formal, structured knowledge representations. This capability contributes one of the most challenging aspects of KE: the acquisition and formalization of tacit and unstructured knowledge. This ability allows them to formalize that tacit knowledge into explicit forms. Their ability to pre-process data complements traditional Symbolic AI, which relies on structured methods such as logical rules. While symbolic AI provides precision and explainability, LLMs contribute with adaptability and scalability.

The integration of LLMs into KE process is achieved by using the logical-linguistic capabilities of LLMs in each step of the CommonKADS protocol as a tool available to the knowledge engineer to speed up the work and have more control over the soundness and completeness of the elicitation process. The objective of the study is therefore to qualify the various steps of KE processes in terms of the structure and quality of the input knowledge and of the output knowledge. Furthermore, for each step the knowledge engineer's interaction protocols with the LLM should be defined. This interaction, in turn, will be studied through a prompt engineering process examining diverse prompt techniques such as few-shot prompting or chain-of-thought-prompting [15].

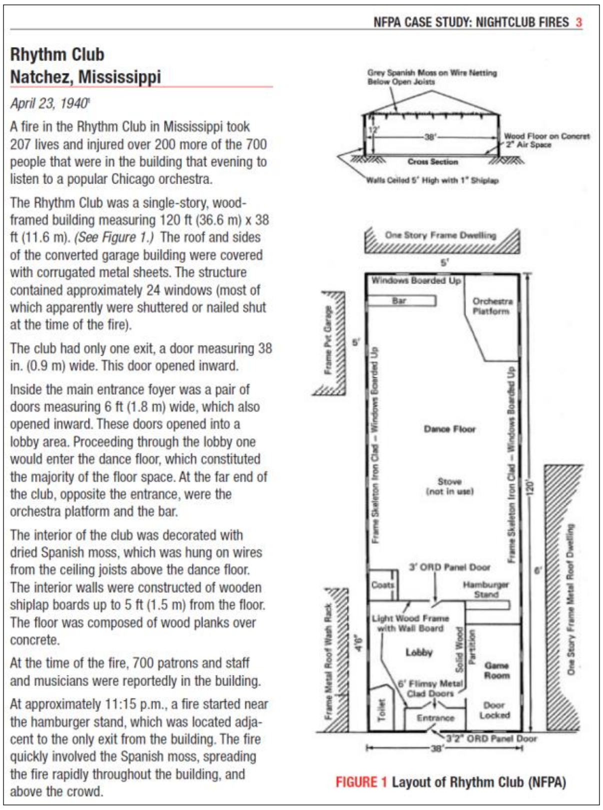
This research contribution presents a phase of development the AI-in-the-loop approach that was presented in Durmus et al. [12], demonstrated the integration of LLMs such as GPT-4 in the development of decision support systems in fire safety planning. These developments benefit from the capabilities of AI to adaptively manage and integrate domain knowledge in dynamic building environments, thus improving operational reliability and decision accuracy.

In the following section annotated examples of the main steps will be provided. The proposed results are the outcomes of a work in progress.

4 The Use of LLMs in the Practice of Knowledge Engineering: Case Study (NFPA)

In this section, we will exemplify the proposed methodology using a case study from the NFPA Case Study: Nightclub Fires report [16]. This manual contains a series of reports of catastrophic fire events described by domain experts. These reports represent a good approximation of what an interview with a domain expert might be. The selection of the NFPA nightclub fire incidents, in particular, was based on the availability of concise and comprehensive documentation that fits well with the LLM's ability to process and analyze texts effectively in terms of text length and complexity. The discussion will be limited to one case study and to the first two phases of the KE process as they share a conceptual level of analysis. The formalization phase, which, on the contrary, involves syntactic aspects, requires specific attention and will be addressed in future works of this research.

As a case study for this research, a scenario from NFPA nightclub fire reports has been selected (Figure 1). Each step for the both phases was firstly done by an expert by manual reading and extracting, then we submitted the same fire event narrative to GPT-4 to do same task and assess its responses.



NFPA CASE STUDY: NIGHTCLUB FIRES 3

Rhythm Club
Natchez, Mississippi

April 23, 1940

A fire in the Rhythm Club in Mississippi took 207 lives and injured over 200 more of the 700 people that were in the building that evening to listen to a popular Chicago orchestra.

The Rhythm Club was a single-story, wood-framed building measuring 120 ft (36.6 m) x 38 ft (11.6 m). *(See Figure 1.)* The roof and sides of the converted garage building were covered with corrugated metal sheets. The structure contained approximately 24 windows (most of which apparently were shuttered or nailed shut at the time of the fire).

The club had only one exit, a door measuring 38 in. (0.9 m) wide. This door opened inward.

Inside the main entrance foyer was a pair of doors measuring 6 ft (1.8 m) wide, which also opened inward. These doors opened into a lobby area. Proceeding through the lobby one would enter the dance floor, which constituted the majority of the floor space. At the far end of the club, opposite the entrance, were the orchestra platform and the bar.

The interior of the club was decorated with dried Spanish moss, which was hung on wires from the ceiling joists above the dance floor. The interior walls were constructed of wooden shiplap boards up to 5 ft (1.5 m) from the floor. The floor was composed of wood planks over concrete.

At the time of the fire, 700 patrons and staff and musicians were reportedly in the building.

At approximately 11:15 p.m., a fire started near the hamburger stand, which was located adjacent to the only exit from the building. The fire quickly involved the Spanish moss, spreading the fire rapidly throughout the building, and above the crowd.

FIGURE 1 Layout of Rhythm Club (NFPA)

Figure 1: Selected case study from NFPA report

4.1 Conceptualization

The conceptualization phase is the foundation of knowledge engineering and focuses on understanding the domain and structuring knowledge in an informal, human-readable format. This phase increases the clarity and accessibility of domain knowledge through natural language to effectively differentiate between static and dynamic aspects of fire events. This type of structuring helps facilitate the detailed analysis of complex scenarios such as a fire emergency. The goal at this stage is to identify and categorize the critical elements of the scenario and identify key issues. In practice, as shown in the Rhythm Club fire case study, information is organized into two structured categories: context arguments (Table 1) and event arguments (Table 2). This categorization abstracts static building characteristics and dynamic aspects of the fire event and thus provides a framework for systematic analysis. Below, a detailed breakdown presents a clear comparison between the insights extracted by a human expert and GPT-4 and demonstrates the practical application and effectiveness of this approach in real-world scenarios.

a) *Context Arguments*: capture the static, descriptive aspects of the scenario, such as the building's structure, layout, and materials.

b) *Event Arguments:* describe the dynamic elements of the fire incident, including the sequence of events, fire propagation, and occupant behaviour during the emergency.

Table 1: Comparison of the context arguments extracted by human expert vs GPT-4

a) Context Arguments	
HUMAN EXPERT	**GPT-4**
	Prompt: *please extract the arguments in following groups: 1.context arguments, 2. event arguments. List the arguments in bullet points.*
The Rhythm Club was a single-story, wood-framed building measuring 120 ft (36.6 m) x 38 ft (11.6 m).	The Rhythm Club was a single-story, wood-framed building measuring 120 ft by 38 ft.
The structure contained approximately 24 windows (most of which apparently were shuttered or nailed shut at the time of the fire).	It had approximately 24 windows, most of which were shuttered or nailed shut at the time of the fire.
The club had only one exit, a door measuring 38 in. (0.9 m) wide.	There was only one exit from the club, a door measuring 38 inches wide, which opened inward.
This door opened inward.	
Inside the main entrance foyer was a pair of doors measuring 6 ft (1.8 m) wide, which also opened inward.	Inside the main entrance foyer, a pair of doors measuring 6 ft wide also opened inward, leading into a lobby area.
These doors opened into a lobby area.	
The floor was composed of wood planks over concrete.	The floor was composed of wood planks over concrete.
(the list goes on…)	*(the list goes on…)**

*In this phase of context argument extraction, domain experts extracted total 12 arguments and ChatGPT-4 extracted 10 arguments. The accuracy of these results was evaluated using the F1 scoring method [17], and the outcome was 0.90, indicating a highly successful performance.

Table 2: Comparison of the event arguments extracted by human expert vs GPT-4

b) Event Arguments	
At the time of the fire, 700 patrons and staff and musicians were reportedly in the building.	On the night of the fire, 700 patrons and staff, including musicians, were in the building.
At approximately 11:15 p.m., a fire started near the hamburger stand, which was located adjacent to the only exit from the building.	The fire started near a hamburger stand located adjacent to the only exit from the building at approximately 11:15 p.m.
The fire quickly involved the Spanish moss, spreading the fire rapidly throughout the building, and above the crowd.	The fire quickly involved the Spanish moss, spreading rapidly throughout the building and above the crowd.
Patrons in the vicinity of the entrance were able to escape.	Patrons near the entrance managed to escape, but those beyond the lobby area were trapped by the fire.
(the list goes on…)	*(the list goes on…)**

*In this phase of event argument extraction, domain experts extracted total 10 arguments and ChatGPT-4 extracted 9 arguments. The accuracy of these results was evaluated using the F1 scoring method, and the outcome was 0.94, indicating a highly successful performance.

4.2 Operationalization

The operationalization phase focuses on converting conceptual knowledge into structured elements, such as triples, to form a preliminary event ontology structure. This structure includes annotated classes, objects, and relationships in order to build a functional ontology suitable to solve problems.

This phase is conceptually complex because the expert, in order to formulate effective problem-solving strategies by means of rule sets, uses a significant amount of general domain knowledge and introduces considerations and conceptualizations that are not present in the description of the initial event.

4.2.1. Extracting Triples and Ontology Elements

By using previously extracted arguments and the 2D layout of the nightclub, triples were extracted to represent objects, relationships, and events.

Table 3: Comparison of the triples extracted by human expert vs GPT-4

HUMAN EXPERT	**GPT-4**
a) Triples From Arguments	
	Prompt: *From these arguments, please extract triples in subject predicate object form. examples: 1. (Rhythm_Club instance-of SINGLE_STORY_BUILDING) 2. (rc_windows instance-of SET) 3. (rc_exit instance-of DOOR) 4.(rc_entrance_foyer instance-of SPACE) 5. (Rhythm_Club structure wood-framed)*
(*Rhythm_Club* **instance-of** SINGLE_STORY_BUILDING)	(Rhythm_Club instance-of SINGLE_STORY_BUILDING)
(*rc_windows* **instance-of** SET)	(rc_roof_cover material corrugated_metal_sheets)
(rc_entrance_foyer **instance-of** SPACE)	(rc_walls_cover material corrugated_metal_sheets)
(rc_windows **instance-of** WINDOW)	(rc_windows instance-of SET)
(rc_windows **number-of** 24)	(rc_windows quantity 24)
(rc_windows ***status*** shuttered)	(rc_windows status shuttered)
(rc_windows ***status*** nailed-shut)	(rc_windows status nailed_shut)
(rc_exit **instance-of** DOOR)	(rc_exit instance-of DOOR)
(List goes on…)	*(List goes on…)**

*In this phase of triples extraction from the previous arguments, domain experts extracted 45 related triples and ChatGPT-4 extracted 23 triples. The accuracy of these results was evaluated using the F1 scoring method and the outcome was 0.53, indicating a satisfactory performance.

Table 4: Comparison of the triples extracted from 2D layout by human expert vs GPT-4

b) Triples From 2D Layout (Figure 1)	
	Prompt: *From the given layout, extract structured data in the form of triples (subject, predicate, object). Examples:* *1. (rc_bar adjacent rc_orchestra_platform)* *2. (rc_centre instance-of POSITION)* *3. (rc_dance_floor is-at rc_centre)*
(rc_bar **adjacent** rc_orchestra_platform)	(rc_bar adjacent rc_orchestra_platform)
(rc_centre **instance-of** POSITION)	(rc_lobby_area instance-of SPACE)
(rc_dance_floor **is-at** rc_centre)	(rc_dance_floor is-at rc_centre)
(rc_entrance **lead-to** lobby)	(rc_entrance lead-to rc_lobby_area)
(rc_coats_room is-near rc_entrance)	(rc_coat_area is-close-to rc_entrance)
(rc_metal_clad_doors instance-of DOOR)	(rc_entrance instance-of DOOR)
(rc_stove is-at dance_floor_centre)	(rc_dance_floor contains stove)
(List goes on…)	*(List goes on…)**

*In this phase of triples extraction from the 2D layout image, domain experts extracted 32 related triples and ChatGPT-4 extracted 21 triples. The accuracy of these results was evaluated using the F1 scoring method, the outcome was 0.67, indicating a satisfactory performance.

Table 5: Comparison of the objects and classes from the previously extracted triples by human expert vs GPT-4

HUMAN EXPERT	**GPT-4**
c) Objects and Classes	
	Prompt: *From the triples you extracted above, extract 1- classes and objects, 2- relationships as following example:* *1-Objects and Classes* *BUILDING TYPE: single-story-building* *COMPONENT: Windows, Door, Floor, Ceiling joists,....* *2- Relations* *above, access, adjacent, component, connect, connected-to,...*
BUILDING TYPE: single-story-building COMPONENT: Windows, Door, Floor, Ceiling joists, Interior walls, Wooden shiplap boards, Layout, Stove MATERIAL: Wood planks, Concrete, Corrugated metal sheets, Metal clad, Spanish moss STATUS: Flimsy, Dried, Trapped SPACE: Entrance, Foyer, Dance floor, Lobby, Bar, Hamburger stand, Orchestra platform, Exit, Coat area, Game room, Toilet POSITION: corner, center OCCUPANT: Patrons, Staff, Musicians	BUILDING TYPE: Single-story-building COMPONENT: Windows, Door, Floor, Ceiling joists, Stove, Lobby area doors MATERIAL: Wood planks, Concrete, Corrugated metal sheets SPACE: Dance floor, Lobby area, Entrance, Coat area, Game room, Toilet, Orchestra platform, Bar POSITION: Centre, Far end of building, Front FURNISHING: Stove (not in use), Doors (flimsy metal clad) STATUS: Not in use, Locked, Boarded up*
d) Relations	
above, access, adjacent, component, connect, connected-to, decorated, dimension, dimension-wide, hung-on, instance-of, is-a, is-at, is-close-to, is-isolated-from, is-near, is-next-to, is-opposite, layed-over, lead-to, made-of, number-of, open, part-of, status, structure	adjacent, is-at, lead-to, is-close-to, contains, located-in, located-at, number-of, dimension-wide, type, status**

*In this phase of extraction of the ontological elements from previous triples, domain experts extracted 6 classes and ChatGPT-4 extracted 7 categories. F1 score is resulted 0,87. **For the extraction of relations, domain expert listed 26 relations while GPT listed 11, this resulted as 0,59 in F1 scoring.

4.2.2. Developing System Goals

The goals of the system are derived from the Rhythm Club case study by analyzing the emergency dynamics, supported by OSHA (Occupational Safety and Health Administration) guidelines for fire safety. These goals are designed to guide the system's monitoring, planning, and action modules effectively. For example, the catastrophic consequences of a single exit, the rapid spread of fire through flammable materials, and the need for coordinated evacuation directly informed Goal 01 (monitoring safety data) and Goal 05 (leading occupants to safety). By breaking down the emergency into manageable objectives, the goals align the system's actions with real-world needs and ensure adaptive, effective responses. Goals represent strategies to minimize the impact of emergencies. Derived from case dynamics and aligned with OSHA guidelines, these rules include:

1. **Goal 01:** Continuously collect fire-safety information.
2. **Goal 02:** Collect fire-emergency information.
3. **Goal 03:** Alert rescue teams, occupants, and fire departments.
4. **Goal 04:** Mitigate fire propagation through suppression and isolation.
5. **Goal 05:** Lead occupants out of the building quickly.
6. **Goal 06:** Ensure occupant safety during evacuation.

4.2.3. Monitoring Rules to achieve defined goals

Monitoring involves tracking system states using sensor data and triggering actions based on significant changes. These rules interact with the defined goals by ensuring real-time system responsiveness. This interaction ensures that the system adapts dynamically to evolving conditions during emergencies.

The use of the LLM allows the rule-setting phase to be approached in an iterative form, in which the context is examined as a whole, and all relevant aspects are identified and examined and investigated in parallel. At this stage, it is essential to control the semantics of the language in a precise manner. A pseudo language based on triples was used for this purpose. The LLMs was asked to: "write a rule set that monitor the distribution of the fire on a fire safety management system, by asserting the triple: (fire <position> <intensity>), concerning the rhythm_club dance floor" using the whole set of triples defined so far in the conceptualization phase as the query context." LLMs was required to provide the result in: "if <conditions> the <actions> format using triples to express statements"

The LLM provided a structured outcome, tagging each rule with a statement about the issue of the rule is concerned on. The following set of issues has been identified by GPT:

1. Fire Detection at Specific Position
2. Fire Intensity Threshold
3. Fire Spreading to Opposite Spaces
4. …

…

11. Fire Alert for Boarded Windows
12. Decorative Material Hazard

A partial example of the rule set extracted by GPT follows:

```
1.  Fire Detection at Specific Position
    If (fire <position>
    <intensity>) and (rc_dance_floor is-at
    <position>).
    Then (alert fire_detected_at
    rc_dance_floor).
2.  Fire Intensity Threshold
    If (fire rc_dance_floor
    <intensity>) and (threshold exceeded_by
    <intensity> 5).
    Then (alert high_fire_intensity
    rc_dance_floor) and (dispatch fire_team to
    rc_dance_floor).
3.  Fire Spreading to Adjacent Spaces
    If (fire rc_dance_floor
    <intensity>) and (threshold exceeded_by
    <intensity> 3).
4.  …
…
11. Fire Alert for Boarded Windows
   If (fire rc_dance_floor <intensity>) and
   (rc_windows status boarded_up).
   Then (alert ventilation_restricted) and
   (alert escape_routes_blocked).
12. Decorative Material Hazard
   If (fire rc_dance_floor <intensity>) and
   (threshold exceeded_by <intensity> 2).
   Then (alert high_flammability_material
   rc_dried_spanish_moss).
```

5 Results and Discussion

The integration of GPT-4 into the knowledge engineering process revealed both its advantages and limitations. F1 scoring has applied to quantitatively assess GPT-4's performance, with experts manually extracting lists from the same texts queried to the model. These expert-curated lists as presented on the tables served as ground truth, against which GPT-4's outputs were compared. In the conceptualization phase, GPT-4 demonstrated high accuracy in extracting context and event arguments, achieving F1 scores of 0.90 and 0.94, respectively. However, its ability to identify nuanced relationships remained limited, emphasizing the importance of domain expert involvement. During operationalization, GPT-4 efficiently extracted triples and ontology elements, but its performance was moderate in capturing complex relationships, with F1 scores of 0.53 for narrative-based triples and 0.67 for layout-based triples. The tested capabilities showed a potential for automating rules development and was close in content, structure, and case specificity, which indicated alignment with what the human experts could produce. Furthermore, this reveals that GPT-4 provides contextual understanding for generating contextually logical and relevant rules from the scenarios representing complex events such as a fire emergency. Moreover, this is achieved without domain-specific, pre-trained embeddings, matching expert-made guidelines of OSHA, demonstrating strong generalization capabilities of the LLM. Despite these strengths, LLMs' reliance on probabilistic methods remain as a limitation, requiring expert oversight to address gaps; this points back AI-in-the-loop model (Durmus et al., 2024) for a hybrid human-ai collaboration for such complex tasks as fire safety management.

6 Conclusion

This study explores the integration of Large Language Models (LLMs) into the knowledge engineering (KE) process, focusing on their application on fire safety management through a hybrid approach that combines human expertise with LLM capabilities. It presents an innovative methodology for modelling both formal and tacit knowledge for decision-making purposes, demonstrating a promising and original approach to integrating structured methodologies like CommonKADS, with the adaptive and scalable potential of LLMs.

The case study on NFPA fire incidents highlighted GPT-4's potential to automate labor-intensive tasks, such as data extraction and rule generation, while complementing expert knowledge. The outcomes are twofold: first, it makes the knowledge engineering process more scalable by providing a framework for structuring and operationalizing large volumes of tacit knowledge in a short time frame; second, it offers a tool to address and adapt to unpredictable aspects, such as emergent patterns in dynamic scenarios. While GPT-4 shows strong potential to contribute many steps of the process, its reliance on probabilistic reasoning and LLMs' lack of explainability imply it cannot yet automate the entire workflow. Expert oversight remains crucial to ensure the accountability and reliability of outputs, particularly in high-risk applications like fire safety management.

Future research should focus on automating the formalization phase and its practical applications, includes applying the methodologies and rules developed in this work using tools such as CLIPS to validate the practical implementation of LLMs into hybrid knowledge engineering systems. Expanding case studies across different construction scenarios would evaluate the scalability and generalizability of AI-integrated knowledge systems. These efforts aim to develop robust and efficient decision-support systems for complex and dynamic environments for construction industry.

7 References

[1] X. Zhu, X. Meng, and M. Zhang, 'Application of multiple criteria decision making methods in construction: a systematic literature review', *Journal of Civil Engineering and Management*, vol. 27, no. 6, pp. 372–403, Jul. 2021, doi: 10.3846/jcem.2021.15260.

[2] D. Ashlock, 'Modeling Frameworks for Knowledge Engineering Approaches', *JCNS*, pp. 15–20, Jan. 2022, doi: 10.53759/181X/JCNS202202003.

[3] M. Stefik, *Introduction to Knowledge Systems*. Elsevier, 2014.

[4] G. Schreiber, *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press, 2000.

[5] L. F. Lai, 'A knowledge engineering approach to knowledge management', *Information Sciences*, vol. 177, no. 19, pp. 4072–4094, Oct. 2007, doi: 10.1016/j.ins.2007.02.028.

[6] J. C. Giarratano and G. Riley, *Expert Systems: Principles and Programming*. USA: Brooks/Cole Publishing Co., 1989.

[7] B. P. Allen, L. Stork, and P. Groth, 'Knowledge Engineering using Large Language Models', Oct. 01, 2023, *arXiv*: arXiv:2310.00637. doi: 10.48550/arXiv.2310.00637.

[8] S. Islam *et al.*, 'A comprehensive survey on applications of transformers for deep learning tasks', *Expert Systems with Applications*, vol. 241, p. 122666, May 2024, doi: 10.1016/j.eswa.2023.122666.

[9] M. Chelli *et al.*, 'Hallucination Rates and Reference Accuracy of ChatGPT and Bard for Systematic Reviews: Comparative Analysis', *Journal of Medical Internet Research*, vol. 26, no. 1, p. e53164, May 2024, doi: 10.2196/53164.

[10] Z. Xie, 'Order Matters in Hallucination: Reasoning Order as Benchmark and Reflexive Prompting for Large-Language-Models', Dec. 30, 2024, *arXiv*: arXiv:2408.05093. doi: 10.48550/arXiv.2408.05093.

[11] P. J. Phillips *et al.*, 'Four principles of explainable artificial intelligence', National Institute of Standards and Technology (U.S.), Gaithersburg, MD, NIST IR 8312, Sep. 2021. doi: 10.6028/NIST.IR.8312.

[12] D. Durmus, A. Giretti, O. Ashkenazi, A. Carbonari, and S. Isaac, 'The Role of Large Language Models for Decision Support in Fire Safety Planning', International Association on Automation and Robotics in Construction, 2024. doi: 10.22260/ISARC2024/0045.

[13] O. Ashkenazi, S. Isaac, A. Giretti, A. Carbonari, and D. Durmus, 'Transforming Building Industry Knowledge Management: A Study on the Role of Large Language Models in Fire Safety Planning', in *Proceedings e report*, P. Capone, V. Getuli, F. Pour Rahimian, N. Dawood, A. Bruttini, and T. Sorbi, Eds., Firenze University Press, 2023. doi: 10.36253/979-12-215-0289-3.73.

[14] A. Ławrynowicz, 'A Knowledge Engineering Primer', Mar. 25, 2024, *arXiv*: arXiv:2305.17196. doi: 10.48550/arXiv.2305.17196.

[15] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, 'A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications', Feb. 05, 2024, *arXiv*: arXiv:2402.07927. doi: 10.48550/arXiv.2402.07927.

[16] R. F. Duval, 'NFPA Case Study: nightclub fires'. National Fire Protection Association, Fire Investigations Department, 2006.

[17] D. Durmus, J. Rožanec, A. Carbonari, A. Giretti, Ž. Turk, and M. Grobelnik, 'Potential of Large Language Models for Ontology Development in Construction Domain', in *Proceedings of the Creative Construction Conference 2024*, Praha, Czech Republic: Budapest University of Technology and Economics, 2024, p. null-null. doi: 10.3311/CCC2024-062.